Copyright © 2013 Tech Science Press CMES, vol.92, no.2, pp.213-239, 2013

A $GL(n,\mathbb{R})$ Differential Algebraic Equation Method for Numerical Differentiation of Noisy Signal

Chein-Shan Liu[1], **Satya N. Atluri**[2]

Abstract: We show that the problem "real-time numerical differentiation" of a given noisy signal in time, by supplementing a compensated controller in the second-order robust exact differentiator, the tracking differentiator or the continuous hybrid differentiator, can be viewed as a set of differential algebraic equations (DAEs) to enhance a precise tracking of the given noisy signal. Thus, we are able to solve the highly ill-posed problem of numerical differentiation of noisy signal by using the Lie-group differential algebraic differentiators (LGDADs) based on the Lie-group $GL(n,\mathbb{R})$, whose accuracy and tracking performance are better than before. The "index-two" differentiators (ITDs), which do not need any parameter in the governing equations, can efficiently depress the chattering, and also improve the dynamical performance under noise. The tracking results obtained by the LGDADs and the ITDs are quite promising.

Keywords: Differential algebraic equations, Lie-group differential algebraic differentiator (LGDAD), Numerical differentiation, Robust exact differentiator, Tracking differentiator, Index-two differentiator (ITD)

1 Introduction

In this paper we cast the problem of numerically finding the first-order and the second-order derivatives of a given noisy signal in time, as a set of time-dependent differential algebraic equations (DAEs). The DAEs govern the evolution of $n+m$ variables $x_i,\ i=1,\ldots,n$ and $y_j,\ j=1,\ldots,m$ with n nonlinear ordinary differential equations (ODEs), and m nonlinear algebraic equations (NAEs):

$$\dot{\mathbf{x}}=\mathbf{f}(\mathbf{x},\mathbf{y},t),\ \ \mathbf{x}(0)=\mathbf{x}_0,\ \ t\in\mathbb{R},\ \ \mathbf{x}\in\mathbb{R}^n,\ \ \mathbf{y}\in\mathbb{R}^m, \tag{1}$$

$$\mathbf{F}(\mathbf{x},\mathbf{y},t)=\mathbf{0},\ \ \mathbf{F}\in\mathbb{R}^m. \tag{2}$$

[1] Department of Civil Engineering, National Taiwan University, Taipei, Taiwan.
E-mail: liucs@ntu.edu.tw
[2] Center for Aerospace Research & Education, University of California, Irvine.

Usually, n is larger than m. When $m = 0$ the DAEs reduce to the ODEs.

The differentiation of "signals", including those with noise, is an old and well-known problem. The problem of numerical differentiation of noisy data is seriously ill-posed: small changes of the data may result in large errors of the derivatives [Liu and Atluri (2009)]. In many applications, it may be necessary to calculate the derivatives of a set of measured data in real time. The first-order and second-order differentials of a given signal $f(t)$ in real time can be modelled by the following "index-three" differential algebraic equations (DAEs):

$$\begin{aligned} &\dot{x}_1(t) = x_2(t), \\ &\dot{x}_2(t) = x_3(t), \\ &x_1(t) - f(t) = 0. \end{aligned} \tag{3}$$

The exact solutions to the above DAEs are obviously $x_1(t) = f(t)$, $x_2(t) = \dot{f}(t)$, and $x_3 = \ddot{f}(t)$; however, we note that the differential operators are highly ill-posed, and one usually cannot obtain the correct solutions by $x_2(t) = \dot{f}(t)$, and $x_3 = \ddot{f}(t)$, when the signal data $f(t)$ are polluted by a random noise. Moreover, the above DAEs have "index-three", and it is very difficult to solve this DAEs by a numerical method.

To mimic the really measured data of $f(t)$ we may impose a random noise on $f(t)$ with an intensity σ. When we require the output x_1 of Eq. (3) to track indeed a noisy signal with $\hat{f}(t_i) = f(t_i) + \sigma R(i)$ where t_i is a sampling time and $R(i)$ is a white noise, we can investigate the tracking errors of displacement and velocity by

$$e_1(t) = x_1(t) - \hat{f}(t), \;\; e_2(t) = x_2(t) - \dot{\hat{f}}(t).$$

Then we can obtain the following dynamics of tracking errors:

$$\dot{e}_1(t) = e_2(t), \;\; \dot{e}_2(t) = x_3(t) - \ddot{\hat{f}}(t).$$

Before the design of a suitable controller $x_3(t)$ to reduce the errors e_1 and e_2, we ask if we can have a correct information about $\dot{\hat{f}}(t)$ and $\ddot{\hat{f}}(t)$, which are extracted from a measured data $\hat{f}(t)$ which is itself being polluted by a random noise. In Section 5 we will seek this information by a different approach than given in Eq. (3).

We consider a second-order ordinary differential equation (ODE) for describing a forced vibration of a linear system with time-dependent parameters $c(t)$ and $k(t)$:

$$\ddot{\phi}(t) + c(t)\dot{\phi}(t) + k(t)\phi(t) = F(t). \tag{4}$$

If we attempt to recover $F(t)$ from the measured data $\phi(t)$, then we are required to compute $\dot{\phi}(t)$ and $\ddot{\phi}(t)$. On the other hand, the damping coefficient $c(t)$ is very

difficult to be measured directly; however, if we can perform the differentials $\dot{\phi}(t)$ and $\ddot{\phi}(t)$ from the measured data $\phi(t)$, then we can estimate $c(t)$.

In addition to its applications in the inverse vibration problems in engineering, numerical differentiation is commonly used in control algorithms to design a controller, where rapidly and accurately computing the velocity and acceleration of a moving target is very important in the controlled system with correct and timely performance. As a consequence, a stable and time-saving differentiator is significant [Ramm and Smirnova (2001); Ahn, Choi and Ramm (2006)]. In the field of control theory the approach by considering the time-differential as a tracking problem to a desired trajectory has spurred many researches, such as the high-gain based differentiator [Dabroom and Khalil (1997)], the linear time-derivative tracker [Ibrir (2004)], the robust exact differentiator [Levant (1998,2003); Levant and Livne (2012)], the finite-time convergent differentiator [Wang, Chen and Yang (2007); Wang and Lin (2012); Wang and Shirinzadeh (2012)], the continuous hybrid differentiator [Wang and Lin (2011)], as well as the tracking differentiator [Han and Wang (1994); Guo, Han and Xi (2002); Tang, Wu, Wu, Hu and Shen (2009); Xue, Huang and Yang (2010); Guo and Zhao (2011a, 2011b)]. When the differentiation problem is tackled by a tracking problem, it requires a short rising time to track the given signal and its differentials. In order to be a real-time numerical differentiation of the tracking signal, we can insert a compensated control force into these differentiators, which is used to stir the initial point into the desired trajectory.

Besides the robust exact differentiator (RED) [Levant (1998,2003)], the tracking differentiator (TD) was popularly used [Han and Wang (1994), Guo, Han and Xi (2002); Tang, Wu, Wu, Hu and Shen (2009); Su, Zheng, Mueller and Duan (2006)]. In this paper we will introduce an extra controller $u(t)$ into the first equation of the RED equations and solve $u(t)$ by the Lie-group DAE (LGDAE) method. By the same token we will modify the TD and the continuous hybrid differentiator (CHD) by adding a compensated control force in the governing equations and solve it by using the LGDAE.

The remaining parts of this paper are arranged as follows. In Section 2 we give a new form of the nonlinear dynamical system, which allows us to derive an implicit $GL(n,\mathbb{R})$ Lie-group scheme in Section 3. Then together with the Newton method we can develop a Lie-group method (LGDAE) to solve the DAEs in Section 4. The main results by applying the LGDAE to solve the real-time numerical differentiation problems of a given noisy signal in time are presented in Section 5. In this section we introduce two new types "index-two" differentiators, and three novel and simple modifications of the robust exact differentiator (RED), tracking differentiator (TD), and continuous hybrid differentiator (CHD), which are collected as the so-called Lie-group differential algebraic differentiators (LGDADs). Finally,

we draw some conclusions in Section 6.

2 The $GL(n,\mathbb{R})$ structure of the differential equations system

Lie-group is a differentiable manifold, which is endowed with a group structure that is compatible with the underlying topology of the manifold. The Lie-group solver can provide a better algorithm that retains the orbit generated from numerical solution on the manifold which is associated with the Lie-group.

The general linear group is a Lie group, whose manifold is an open subset $GL(n,\mathbb{R}) := \{\mathbf{G} \in \mathbb{R}^{n\times n} | \det \mathbf{G} \neq 0\}$ of the linear space of all $n \times n$ non-singular matrices. Thus, $GL(n,\mathbb{R})$ is also an $n \times n$-dimensional manifold. The group composition is given by the matrix multiplication.

The general linear group $GL(n,\mathbb{R})$ gives uniquely a real Lie-algebra $gl(n,\mathbb{R})$. Consider a one-parameter subgroup $\mathbf{G}(t)$, $t \in \mathbb{R}$, of the general linear group $GL(n,\mathbb{R})$, which is a curve passing through the group identity at $t = 0$,

$$\mathbf{G}(0) = \mathbf{I}_n, \tag{5}$$

and which operates from the left on the n-dimensional Euclidean space $\mathbb{R}^n$, resulting in a congruence of curves in $\mathbb{R}^n$,

$$\mathbf{X}(t) = \mathbf{G}(t)\mathbf{X}(0) = [\mathbf{G}(t)\mathbf{G}^{-1}(t_0)]\mathbf{X}(t_0), \;\; t_0, t \in \mathbb{R}. \tag{6}$$

Owing to the closure property of the Lie group, $\mathbf{G}(t)\mathbf{G}^{-1}(t_0)$ also belongs to $GL(n,\mathbb{R})$. When t_0 is put very close to t, $\mathbf{G}(t)\mathbf{G}^{-1}(t_0)$ is very close to the identity $\mathbf{I}_n$. Moreover,

$$\mathbf{A}(t) := \frac{\partial}{\partial t}[\mathbf{G}(t)\mathbf{G}^{-1}(t_0)]\Big|_{t_0=t} = \dot{\mathbf{G}}(t)\mathbf{G}^{-1}(t) \tag{7}$$

defines a string of tangent vectors on the tangent space at the group identity of the group manifold, more precisely, a continuously singly parametrized series of one-dimensional sub-algebra of the real Lie algebra $gl(n,\mathbb{R})$.

Differentiating Eq. (6), setting $t_0 = t$ and then using Eq. (7) yields

$$\dot{\mathbf{X}}(t) = \mathbf{A}(t)\mathbf{X}(t). \tag{8}$$

The flow generated by the vector field $\mathbf{AX}$ with such an $\mathbf{A}(t) \in gl(n,\mathbb{R})$ is the congruence of curves resulting from solving the linear ODEs system (8). Essentially, it is very important that when one wants to establish a local coordinate on a smooth manifold on which a finite dimensional Lie-group of transformation is acting, the solution of the differential equations system is necessary [Liu (2001,2007)].

Here we give a new form of the dynamics in Eq. (1) from the $GL(n,\mathbb{R})$ Lie-group structure. In order to fit the form in Eq. (8), the vector field $\mathbf{f}$ on the right-hand side of Eq. (1) can be written as

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x}, \tag{9}$$

where

$$\mathbf{A} = \frac{\mathbf{f}}{\|\mathbf{x}\|} \otimes \frac{\mathbf{x}}{\|\mathbf{x}\|} \tag{10}$$

is the coefficient matrix. Here $\mathbf{u} \otimes \mathbf{y}$ denotes the dyadic operation of $\mathbf{u}$ and $\mathbf{y}$, i.e., $(\mathbf{u} \otimes \mathbf{y})\mathbf{z} = \mathbf{y} \cdot \mathbf{z}\mathbf{u}$.

Because the coefficient matrix $\mathbf{A}$ is well-defined for $\|\mathbf{x}\| > 0$, the Lie-group element $\mathbf{G}$ generated from the above dynamical system (9) with $\dot{\mathbf{G}} = \mathbf{A}\mathbf{G}$ satisfies $\det \mathbf{G}(t) \neq 0$, such that $\mathbf{G} \in GL(n,\mathbb{R})$.

Liu (2013a) was the first to find the essential form in Eq. (9) for nonlinear ODEs, and developed a very effective Lie-group $GL(n,\mathbb{R})$ preserving scheme to solve ODEs by only assuming that $\mathbf{f}/\|\mathbf{x}\|$ is a constant vector. Then, Liu (2013b) developed a Lie-group $GL(n,\mathbb{R})$ preserving scheme to solve ODEs by assuming that both $\mathbf{f}/\|\mathbf{x}\|$ and $\mathbf{x}/\|\mathbf{x}\|$ are constant vectors.

3 An implicit $GL(n,\mathbb{R})$ Lie-group scheme

Eq. (9) is a new starting point for the development of the Lie-group $GL(n,\mathbb{R})$ scheme. In order to develop a numerical scheme from Eqs. (9) and (10), we suppose that the coefficient matrix $\mathbf{A}$ is constant with

$$\mathbf{a} = \frac{\bar{\mathbf{f}}}{\|\bar{\mathbf{x}}\|}, \quad \mathbf{b} = \frac{\bar{\mathbf{x}}}{\|\bar{\mathbf{x}}\|} \tag{11}$$

being two constant vectors, which can be obtained by taking the values of $\mathbf{f}$ and $\mathbf{x}$ at a suitable mid-point of $\bar{t} \in [t_0 = 0, t]$, where $t \leq t_0 + h$ and h is a small time stepsize. Thus from Eqs. (9) and (10) we have

$$\dot{\mathbf{x}} = \mathbf{b} \cdot \mathbf{x}\mathbf{a}. \tag{12}$$

Let

$$w = \mathbf{b} \cdot \mathbf{x}, \tag{13}$$

and Eq. (12) becomes

$$\dot{\mathbf{x}} = w\mathbf{a}. \tag{14}$$

At the same time, from the above two equations we can derive the following ODE for w:

$$\dot{w} = cw, \tag{15}$$

where

$$c = \mathbf{a} \cdot \mathbf{b} \tag{16}$$

is viewed as a constant scalar in the time interval of $t \in [t_0, t_0 + h]$. Thus we have

$$w(t) = w_0 \exp(ct), \quad t \in [t_0, t_0 + h], \tag{17}$$

where $w_0 = \mathbf{b} \cdot \mathbf{x}_0$.

Inserting Eq. (17) for $w(t)$ into Eq. (14) and integrating the resultant equation we can obtain

$$\mathbf{x}(t) = [\mathbf{I}_n + \eta(t)\mathbf{a}\mathbf{b}^{\mathrm{T}}]\mathbf{x}_0, \tag{18}$$

where $\mathbf{x}_0$ is the initial value of $\mathbf{x}$ at an initial time $t = t_0 = 0$, and

$$\eta(t) = \frac{e^{ct} - 1}{c}. \tag{19}$$

Let $\mathbf{G}(t)$ be the coefficient matrix before $\mathbf{x}_0$ in Eq. (18),

$$\mathbf{G} = \mathbf{I}_n + \eta(t)\mathbf{a}\mathbf{b}^{\mathrm{T}}, \tag{20}$$

and we can prove that

$$\det \mathbf{G} = e^{ct} > 0, \tag{21}$$

which means that $\mathbf{G}$ is a Lie-group element of $GL(n, \mathbb{R})$. $\mathbf{G}$ has the following properties.

(1) If $\mathbf{a} \cdot \mathbf{b} \neq 0$, then $\mathbf{G}$ has n linearly independent eigenvectors, which are composed of $\mathbf{a}$ and $n-1$ arbitrarily linearly independent vectors in $\mathbf{b}^{\perp}$, where $\mathbf{b}^{\perp}$ is an $(n-1)$-dimensional subspace being orthogonal to $\mathbf{b}$. If $\mathbf{a} \cdot \mathbf{b} = 0$, then $\mathbf{G}$ only has $n-1$ linearly independent eigenvectors, which are composed of $n-1$ arbitrarily linearly independent vectors in $\mathbf{b}^{\perp}$.

Proof: First we choose a set of bases $\mathbf{u}_i,\ i = 1, \ldots, n-1$ in $\mathbf{b}^{\perp}$, then one has

$$\mathbf{G}\mathbf{u}_i = \mathbf{u}_i,\ i = 1, \ldots, n-1, \tag{22}$$

which means that $\mathbf{u}_i,\ i=1,\ldots,n-1$ are the eigenvectors of $\mathbf{G}$ corresponding to the eigenvalue $\lambda=1$.

If $\mathbf{a}\notin\mathbf{b}^{\perp}$, then we have

$$\mathbf{G}\mathbf{a}=[1+\eta\,\mathbf{a}\cdot\mathbf{b}]\mathbf{a}, \tag{23}$$

which means that $\mathbf{a}$ is also the eigenvector of $\mathbf{G}$ with eigenvalue $\lambda=1+\eta\,\mathbf{a}\cdot\mathbf{b}$.

If $\mathbf{a}\in\mathbf{b}^{\perp}$, then any eigenvector $\mathbf{u}$ of $\mathbf{G}$ must be $\mathbf{u}\in\mathbf{b}^{\perp}$. By the assumption that $\mathbf{u}$ is an eigenvector of $\mathbf{G}$ we have

$$\mathbf{G}\mathbf{u}=q\mathbf{u}. \tag{24}$$

Then insert Eq. (20) for $\mathbf{G}$ we have

$$(1-q)\mathbf{u}=-\eta\,\mathbf{b}\cdot\mathbf{u}\mathbf{a}. \tag{25}$$

If $q=1$ then $\mathbf{b}\cdot\mathbf{u}=0$, i.e., $\mathbf{u}\in\mathbf{b}^{\perp}$. Suppose that $q\neq 1$. We assume that $\mathbf{u}\notin\mathbf{b}^{\perp}$, and from the above equation we know that $\mathbf{u}$ is collinear with $\mathbf{a}$, which however, contradicts with the premise $\mathbf{a}\in\mathbf{b}^{\perp}$. Thus we must have $\mathbf{u}\in\mathbf{b}^{\perp}$.

(2) The eigenvalues of $\mathbf{G}$. We can take a set of orthogonal vectors $\mathbf{u}_2,\ldots,\mathbf{u}_n$ in $\mathbf{b}^{\perp}$. Then we can construct an orthogonal matrix $\mathbf{U}$ by

$$\mathbf{U}=(\mathbf{b},\mathbf{u}_2,\ldots,\mathbf{u}_n)\,. \tag{26}$$

By using

$$\mathbf{a}\mathbf{b}^{\mathrm{T}}\mathbf{U}=\mathbf{U}\begin{bmatrix}\mathbf{b}^{\mathrm{T}}\mathbf{a} & 0 & \ldots & 0\\ 0 & 0 & \ldots & 0\\ 0 & \vdots & \vdots & 0\\ 0 & \ldots & \ldots & 0\end{bmatrix}, \tag{27}$$

we can obtain

$$\mathbf{G}\mathbf{U}=\mathbf{U}\left(\mathbf{I}+\eta\begin{bmatrix}\mathbf{b}^{\mathrm{T}}\mathbf{a} & 0 & \ldots & 0\\ 0 & 0 & \ldots & 0\\ \vdots & \vdots & \vdots & 0\\ 0 & \ldots & \ldots & 0\end{bmatrix}\right), \tag{28}$$

that is,

$$\mathbf{U}^{\mathrm{T}}\mathbf{G}\mathbf{U}=\begin{bmatrix}1+\eta\mathbf{b}^{\mathrm{T}}\mathbf{a} & 0 & \ldots & 0\\ 0 & 1 & \ldots & 0\\ \vdots & \vdots & \vdots & 0\\ 0 & \ldots & \ldots & 1\end{bmatrix}. \tag{29}$$

Thus the eigenvalues and determinant of $\mathbf{G}$ are, respectively,

$$\{\lambda(\mathbf{G})\} = \{1+\eta\mathbf{a}\cdot\mathbf{b}, 1, \ldots, 1\}, \tag{30}$$

$$\det\mathbf{G} = 1+\eta\mathbf{a}\cdot\mathbf{b} = e^{ct} > 0. \tag{31}$$

Hence, we have proven Eq. (21).

Accordingly, we can develop the following implicit scheme for solving the ODEs in Eq. (1):
(i) Give $0 \le \theta \le 1$.
(ii) Give an initial $\mathbf{x}_0$ at an initial time $t = t_0$ and a time stepsize h.
(iii) For $k = 0, 1, \ldots$, we repeat the following computations to a specified terminal time $t = t_f$:

$$\mathbf{x}_{k+1} = \mathbf{x}_k + h\mathbf{f}_k. \tag{32}$$

With the above $\mathbf{x}_{k+1}$ generated from an Euler step as an initial guess we iteratively solve the new $\mathbf{x}_{k+1}$ by

$$\begin{aligned}
&\bar{\mathbf{x}}_k = (1-\theta)\mathbf{x}_k + \theta\mathbf{x}_{k+1},\\
&\mathbf{a}_k = \frac{\bar{\mathbf{f}}_k}{\|\bar{\mathbf{x}}_k\|} = \frac{\mathbf{f}(\bar{\mathbf{x}}_k, \mathbf{y}_k, t_k+\theta h)}{\|\bar{\mathbf{x}}_k\|},\\
&\mathbf{b}_k = \frac{\bar{\mathbf{x}}_k}{\|\bar{\mathbf{x}}_k\|},\\
&c_k = \mathbf{a}_k\cdot\mathbf{b}_k,\\
&\eta_k = \frac{\exp(c_k h)-1}{c_k},\\
&\mathbf{z}_{k+1} = \mathbf{x}_k + \eta_k\mathbf{b}_k\cdot\mathbf{x}_k\mathbf{a}_k.
\end{aligned} \tag{33}$$

If $\mathbf{z}_{k+1}$ converges according to a given stopping criterion, such that,

$$\|\mathbf{z}_{k+1} - \mathbf{x}_{k+1}\| < \varepsilon_2, \tag{34}$$

then go to (iii) to the next time step; otherwise, let $\mathbf{x}_{k+1} = \mathbf{z}_{k+1}$ and go to Eq. (33). In all the computations given below we will fix $\theta = 1/2$. In the above $\mathbf{y}_k$ is viewed as a constant vector within a small time step. In order to determine $\mathbf{y}_k$ the present Lie-group scheme is coupled with the following Newton scheme.

4 Newton iterative scheme for DAEs

Now, we turn our attention to the DAEs defined in Eqs. (1) and (2). Within a small time step we can suppose that the variables $y_j,\ j = 1, \ldots, m$ are constant in that

interval of $t_k < t < t_{k+1}$. We give an initial guess of $y_j,\ j = 1,\ldots,m$, and insert them into Eq. (1). Then we apply the above implicit scheme to find the next $\mathbf{x}_{k+1}$, supposing that $\mathbf{x}_k$ is already obtained in the previous time step. When $\mathbf{x}_{k+1}$ are available we insert them into Eq. (2), and then apply the Newton iterative scheme to solve $\mathbf{y}_{k+1}$ by

$$\mathbf{y}_{k+1}^{\ell+1} = \mathbf{y}_{k+1}^{\ell} - \mathbf{B}^{-1}\mathbf{F}(\mathbf{x}_{k+1}, \mathbf{y}_{k+1}^{\ell}, t_{k+1}), \tag{35}$$

until the following convergece criterion is satisfied:

$$\|\mathbf{y}_{k+1}^{\ell+1} - \mathbf{y}_{k+1}^{\ell}\| < \varepsilon_1. \tag{36}$$

In the above the component B_{ij} of the Jacobian matrix $\mathbf{B}$ is given by $\partial F_i/\partial y_j$.

The numerical process as a combination of the Lie-group method based on $GL(n,\mathbb{R})$ and the Newton method to solve the DAEs in Eqs. (1) and (2) is called the Lie-group DAE (LGDAE) method.

5 Numerical differentials

Now we are ready to apply the above LGDAE to solve the numerical problem of a real-time numerical differentiation of a given noisy signal in time. The main idea is modified the proposed differentiators into a set of DAEs including an unknown control force $u(t)$, which is then solved by the LGDAE method.

5.1 The first "index-two" differentiator

In Section 1 we have mentioned a naive and very simple approach of the first-order and the second-order differentials of a given function $f(t)$ by Eq. (3), which is however only a definition of derivatives. When we apply the LGDAE to solve Eq. (3) we find that the numerical results of differentials are poor when time is over 1 sec or when the given function is polluted by noise. In order to obtain a reasonable result the time stepsize cannot be too small. However, Eq. (3) is a prototype of the so-called second-order differentiator to realize the first-order differential of $g(t)$, which is re-formulated as

$$\begin{aligned} \dot{x}(t) &= y(t), \\ \dot{y}(t) &= u(t), \\ y(t) - g(t) &= 0. \end{aligned} \tag{37}$$

Here we view the above equations as a set of DAEs, which has index two, and we search the controller $u(t)$ such that $y(t)$ can track the given data $\hat{g}(t_i)$ where a random noise is imposed on $g(t_i)$ with an intensity σ: $\hat{g}(t_i) = g(t_i) + \sigma R(i)$. Eq. (37) is

an integral-differentiator like the PID [Su, Sun and Duan (2005)] used in the control machine, where $x(t)$ is the integral of the given signal $g(t)$ and $u(t)$ is the differential $\dot{g}(t)$ of the given signal $g(t)$ when $y(t)$ can track the signal $g(t)$. We apply the Lie-group DAE (LGDAE) method to solve the above equations and compare the results with the exact solutions:

$$\int g(t)dt = 2+t+5t^2/2-\cos t,\ \ g(t) = 1+5t+\sin t,\ \ \dot{g}(t) = 5+\cos t$$

in a time interval of $t \in [0,20]$. The differentiator in Eq. (37) is named as an "index-two" differentiator (ITD), which is different from the index-three differentiator in Eq. (3) on the third equation.

We apply the LGDAE method in Sections 3 and 4 to solve $x_1 := x$ and $x_2 := y$ through Eq. (37), and then iteratively solve the constraint in Eq. (37) by the Newton method. For demonstration purposes the numerical processes are given below:
(i) Give $0 \le \theta \le 1$ and an initial guess of u_0.
(ii) Give an initial $\mathbf{x}_0$ at an initial time $t = t_0$ and a time stepsize h.
(iii) For $k = 0, 1, \ldots$, we repeat the following computations to $t = t_f$:

$$\mathbf{x}_{k+1} = \mathbf{x}_k + h\mathbf{f}_k. \tag{38}$$

With the above $\mathbf{x}_{k+1}$ generated from an Euler step as an initial guess we iteratively solve the new $\mathbf{x}_{k+1}$ by

$$\begin{aligned}
&\bar{\mathbf{x}}_k = (1-\theta)\mathbf{x}_k + \theta\mathbf{x}_{k+1},\\
&\mathbf{a}_k = \frac{\bar{\mathbf{f}}_k}{\|\bar{\mathbf{x}}_k\|} = \frac{\mathbf{f}(\bar{\mathbf{x}}_k, t_k+\theta h)}{\|\bar{\mathbf{x}}_k\|},\\
&\mathbf{b}_k = \frac{\bar{\mathbf{x}}_k}{\|\bar{\mathbf{x}}_k\|},\\
&c_k = \mathbf{a}_k \cdot \mathbf{b}_k,\\
&\eta_k = \frac{\exp(c_k h)-1}{c_k},\\
&\mathbf{z}_{k+1} = \mathbf{x}_k + \eta_k \mathbf{b}_k \cdot \mathbf{x}_k \mathbf{a}_k.
\end{aligned} \tag{39}$$

If $\mathbf{z}_{k+1}$ converges according to a given stopping criterion, such that,

$$\|\mathbf{z}_{k+1} - \mathbf{x}_{k+1}\| < \varepsilon_2, \tag{40}$$

then go to (iv); otherwise, let $\mathbf{x}_{k+1} = \mathbf{z}_{k+1}$ and go to Eq. (39).
(iv) For $j = 0, 1, \ldots$, we repeat the following computations:

$$u_{j+1} = u_j - \frac{F_j}{F_j'}, \tag{41}$$

where

$$
\begin{aligned}
F_j &= x_2^{k+1} - g(t_{k+1}),\\
c_k' &= \frac{b_2^k}{\|\bar{\mathbf{x}}_k\|},\\
d_k &= \mathbf{b}_k \cdot \mathbf{x}_k,\\
\eta_k' &= \frac{c_k'[(hc_k - 1)\exp(c_k h) + 1]}{c_k^2},\\
F_j' &= (x_2^{k+1})' = a_2^k d_k \eta_k' + \frac{d_k \eta_k}{\|\bar{\mathbf{x}}_k\|},
\end{aligned} \tag{42}
$$

in which a_2^k and b_2^k are, respectively, the second components of $\mathbf{a}_k$ and $\mathbf{b}_k$. If u_j converges according to

$$|u_{j+1} - u_j| < \varepsilon_1, \tag{43}$$

then go to (iii) with u_j as an initial guess of u for the next time step; otherwise, let $u_j = u_{j+1}$ and go to Eq. (39). In all computations given below we will fix $\theta = 1/2$. In the solution of Eq. (37) by using the LGDAE we fix $h = 0.05$, $\varepsilon_1 = 10^{-6}$, $\varepsilon_2 = 10^{-15}$, and the noise intensity is $\sigma = 0.001$. From Fig. 1 it can be seen that the numerical solutions obtained by the first ITD are close to the exact solutions. The maximum errors for $\int g(t)dt$, $g(t)$ and $\dot{g}(t)$ are, respectively, 1.124×10^{-2}, 9.99×10^{-4}, and 6.397×10^{-2}. Although the ITD is simple, it can lead to rather accurate numerical results. In order to obtain a higher-order differential of a given signal a more complex differentiator is necessary, which is considered below.

5.2 A modification of RED

In the control theory, a known approach of the differential problem is choosing a high quality controller $u(t)$ to track the signal $f(t)$ by $x(t)$ with $\dot{x}(t) = u(t)$ [Golembo, Emelyanov, Utkin and Shubladze(1976)]. A well known robust exact differentiator (RED) was first derived by Levant (1993, 1998) to form a continuous control $u(t)$ for retaining the equalities $s = x(t) - f(t) = 0$ and $\dot{s} = u(t) - \dot{f}(t) = 0$ in time:

$$\dot{x}(t) = y(t) - \alpha|x(t) - f(t)|^{1/2}\text{sign}(x(t) - f(t)), \tag{44}$$

$$\dot{y}(t) = -\beta\,\text{sign}(x(t) - f(t)), \tag{45}$$

where $\alpha > 0$ and $\beta > 0$ are two parameters used to speed up the convergence, and sign is a signum function.

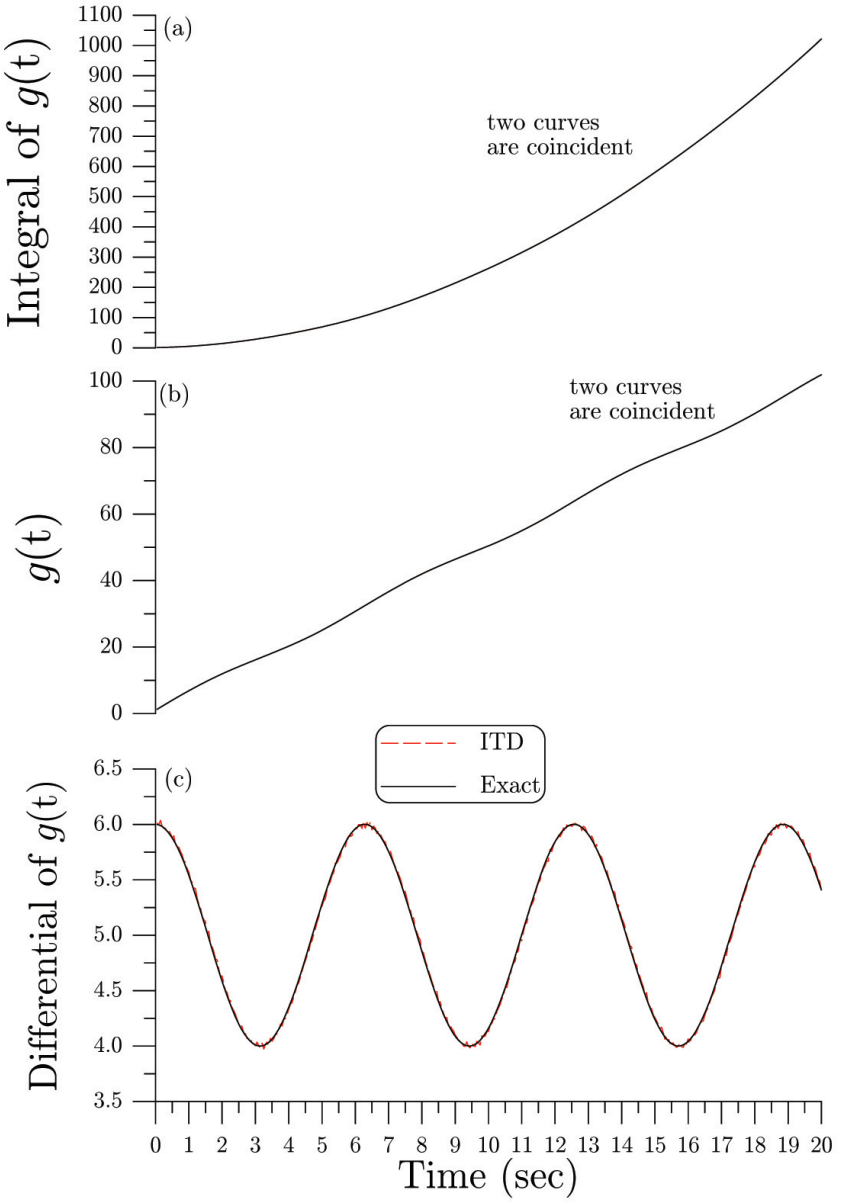

Figure 1: For the first-order differential computed by the first ITD, comparing the numerical and exact solutions of (a) integral of $g(t)$, (b) $g(t)$ and (c) differential of $g(t)$.

Levant (2003) has derived a higher-order differentiator of $f(t)$, of which we only write the following third-order differentiator:

$$\dot{x}(t) = y(t) - \lambda_1 |x(t) - f(t)|^{2/3} \text{sign}(x(t) - f(t)), \tag{46}$$
$$\dot{y}(t) = z(t) - \lambda_2 |y(t) - \dot{x}(t)|^{1/2} \text{sign}(y(t) - \dot{x}(t)), \tag{47}$$
$$\dot{z}(t) = -\lambda_3 \text{sign}(z(t) - \dot{y}(t)), \tag{48}$$

where $\lambda_i > 0,\ i = 1,2,3$ are parameters used to speed up the convergence, and $(x(t), y(t), z(t))$ provides a tracking of the signal and its differentials $(f(t), \dot{f}(t), \ddot{f}(t))$.

Here we modify Eqs. (44) and (45) for the first-order differential of $f(t)$ by

$$\dot{x}(t) = y(t) - \alpha |x(t) - f(t)|^{1/2} \text{sign}(x(t) - f(t)) + u(t), \tag{49}$$
$$\dot{y}(t) = -\beta \text{sign}(y(t) - \dot{x}(t)), \tag{50}$$

where the role of $u(t)$ is used to keep the trajectory of $x(t)$ on the dynamical sliding line:

$$s(t) = x(t) - f(t) = 0. \tag{51}$$

Eqs. (49)-(51) constitute a set of DAEs, and we can apply the LGDAE to solve them. Hence, we name it a Lie-group differential algebraic differentiator (LGDAD). When $u = 0$, Eqs. (49) and (50) are equivalent to Eqs. (44) and (45).

For the first example we test the differential of $f(t) = 5t + \sin t$, and hence, $\dot{f}(t) = 5 + \cos t$. In the solution of Eqs. (49)-(51) by using the LGDAE we fix $h = 0.05$, $\varepsilon_1 = 10^{-5}$, $\varepsilon_2 = 10^{-8}$, $\alpha = 3$ and $\beta = 1$, and the noise intensity is $\sigma = 0.001$. From Fig. 2(a) it can be seen that the numerical solution of LGDAD is close to the exact solution. If we do not consider the compensated control force with $u = 0$ and using the Euler-integration to the above equations (44) and (45) as a differential, as shown in Fig. 2(b) we have a larger error in the numerical differential due to the influence of noise. While the maximum error obtained by RED is 0.19, that obtained by LGDAD is 0.09997, which is is not better than that obtained by the first ITD as shown in Fig. 1(c). We can observe that the chattering phenomenon appears in the differential curve obtained by the RED.

The extra compensated control force used in the LGDAD is shown in Fig. 2(c). It can be seen that the differential obtained by the LGDAD is rather smooth without having a zig-zag, although a random noise is imposed on the differentiator.

Then, we modify Eqs. (46)-(48) for the second-order differential of $f(t)$ by

$$\dot{x}(t) = y(t) - \lambda_1 |x(t) - f(t)|^{2/3} \text{sign}(x(t) - f(t)) + u(t), \tag{52}$$
$$\dot{y}(t) = z(t) - \lambda_2 |y(t) - \dot{x}(t)|^{1/2} \text{sign}(y(t) - \dot{x}(t)), \tag{53}$$
$$\dot{z}(t) = -\lambda_3 \text{sign}(z(t) - \dot{y}(t)). \tag{54}$$

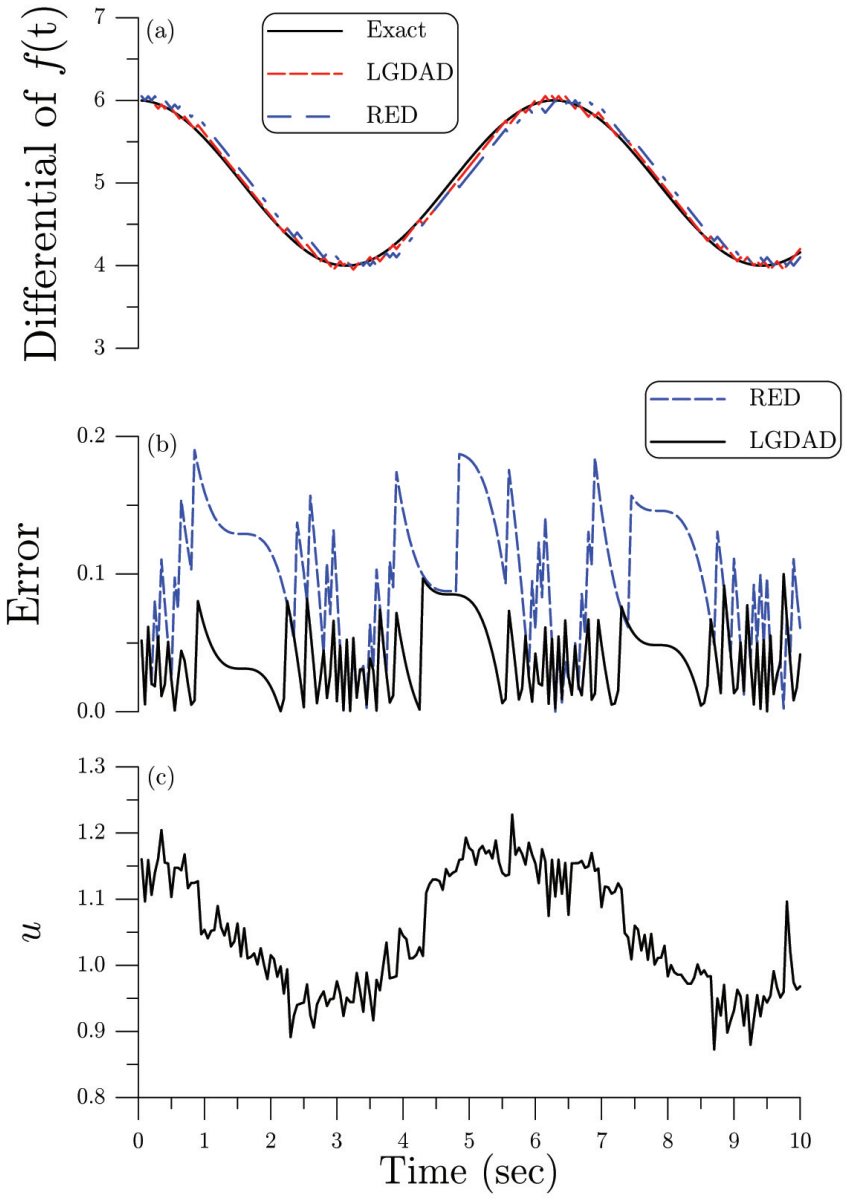

Figure 2: For example of first-order differential, (a) comparing the numerical and exact solutions, (b) the numerical errors, and (c) the control force.

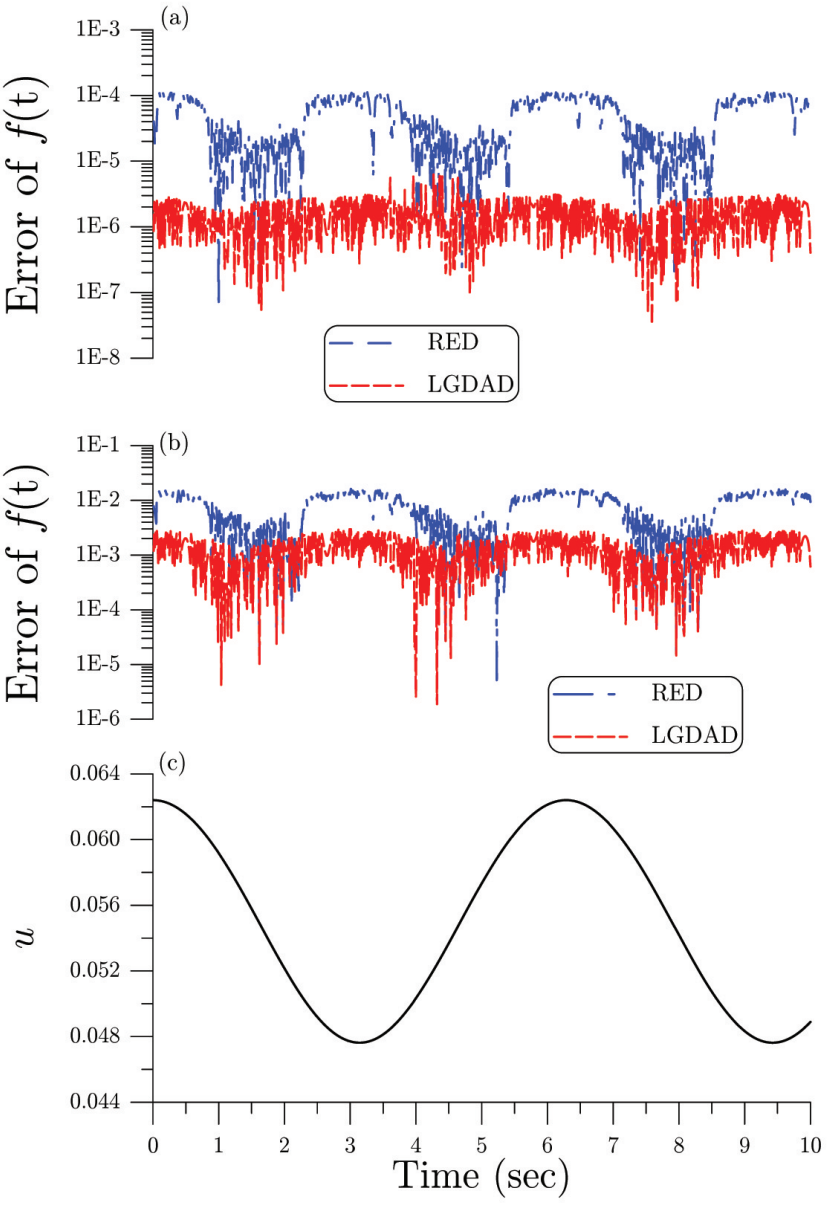

Figure 3: For example of second-order differential, (a) comparing the numerical and exact solutions, (b) the numerical errors, and (c) the control force.

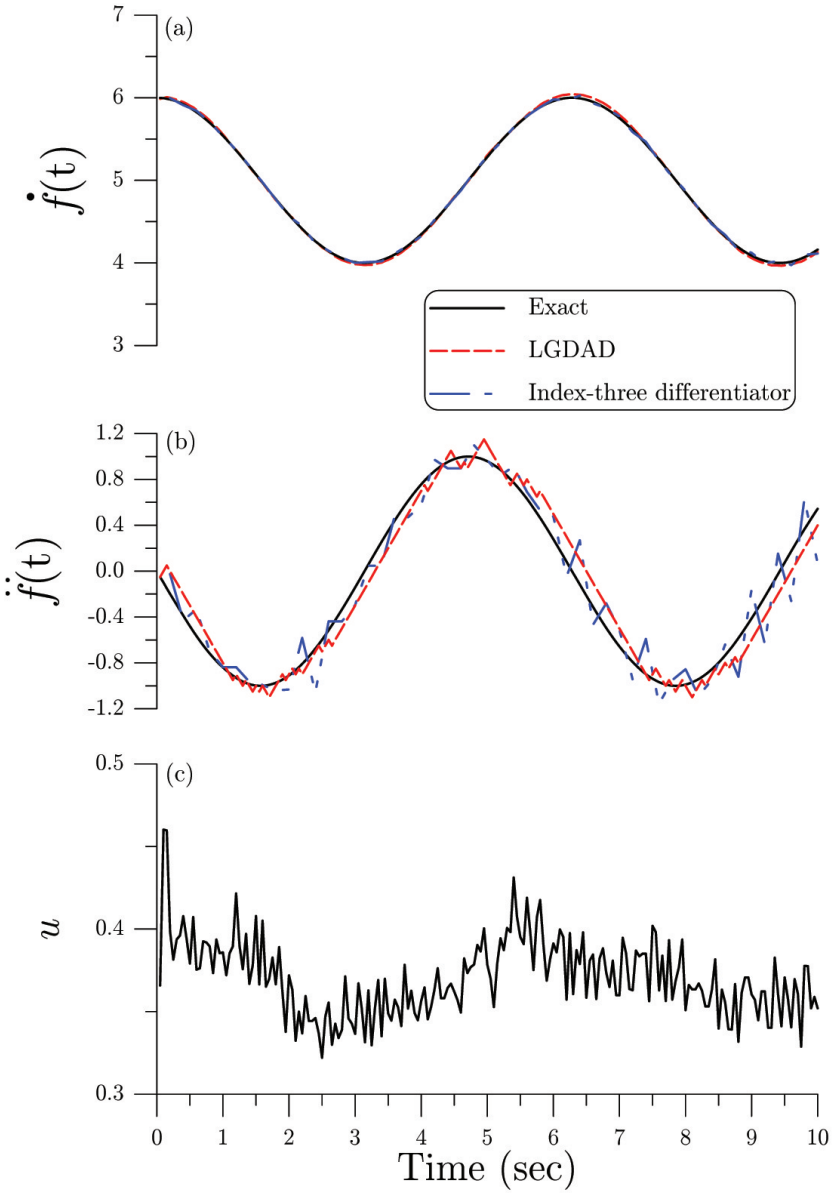

Figure 4: For example of second-order differential under a noise, (a) and (b) comparing the numerical and exact solutions, and (c) the control force of LGDAD.

Eqs. (52)-(54) and (51) constitute a set of DAEs, and we can apply the LGDAE to solve them. Here we have introduced an extra compensated control force $u(t)$ into the first equation of the above equations and solved $u(t)$ by the LGDAE.

For this example we test the first and the second differentials of $f(t) = 5t + \sin t$. By using the LGDAE we fix $h = 0.001$, $\varepsilon_1 = 10^{-11}$, $\varepsilon_2 = 10^{-15}$, $\lambda_1 = 3$, $\lambda_2 = 1.5$, and $\lambda_3 = 1.1$ and the noise intensity is $\sigma = 0$. From Fig. 3 we can observe that the error of $\dot{f}(t)$ obtained by the LGDAD with the maximum error being 6.45×10^{-6} is much smaller with several orders than that without using u. For the LGDAD the error of $\ddot{f}(t)$ is about 3.02×10^{-3}, which is also better than that obtained by the RED with $u = 0$. The extra compensated control force used in the LGDAD is shown in Fig. 3(c).

Under a noise with an intensity $\sigma = 0.001$, we fix $h = 0.05$, $\varepsilon_1 = 10^{-3}$, $\varepsilon_2 = 10^{-7}$, $\lambda_1 = 1.5$, $\lambda_2 = 1$, and $\lambda_3 = 1$. From Figs. 4(a) and 4(b) we can observe that the numerical results are acceptable, where the extra compensated control force used in the LGDAD is shown in Fig. 4(c). While the maximum error of $\dot{f}(t)$ is 0.0464, the maximum error of $\ddot{f}(t)$ is 0.235.

For the purpose of comparison we also use the LGDAE method to solve Eq. (3) under $h = 0.2$, $\varepsilon_1 = 10^{-3}$, and $\varepsilon_2 = 10^{-10}$. This index-three differentiator is good in $\dot{f}(t)$ as shown in Fig. 4(a) by the dashed-dotted line; however, as shown in Fig. 4(b) the result of $\ddot{f}(t)$ is slightly worse than that obtained by the LGDAD. The advantage by using the index-three equation method is that we do not need any parameter in the governing equations. However, when a smaller time stepsize h is adopted the performance of index-three equation method is not so good.

5.3 A modification of TD

For the first-order differential of a given signal $f(t)$, one feasible second-order Tracking Differentiator (TD) can be expressed as [Han and Wang (1994); Su, Sun and Duan (2005); Wang, Zhang and Yan (2010)]:

$$\dot{x}(t) = y(t), \tag{55}$$

$$\dot{y}(t) = -\alpha \text{sign}\left(x(t) - f(t) + \frac{y(t)|y(t)|}{2\alpha}\right). \tag{56}$$

Now we modify the above TD by the following DAE:

$$\dot{x}(t) = y(t) + u(t), \tag{57}$$

$$\dot{y}(t) = u(t) - \alpha \text{sign}\left(x(t) - f(t) + \frac{y(t)|y(t)|}{2\alpha}\right), \tag{58}$$

$$x(t) - f(t) = 0. \tag{59}$$

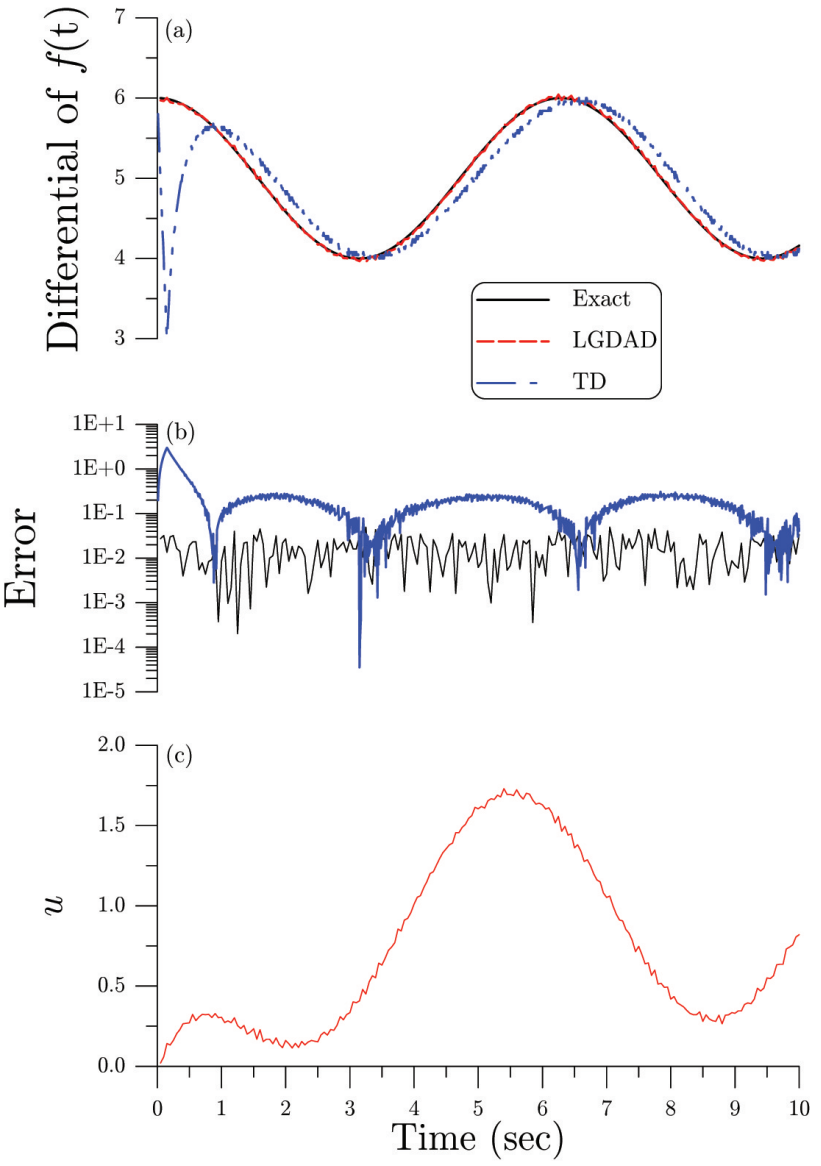

Figure 5: For example of first-order differential, (a) comparing the numerical solutions obtained by LGDAD and TD with exact solution, (b) the numerical errors, and (c) the control force for LGDAD.

Hence, we attempt to seek a suitable controller u such that $x(t)$ can track the signal $f(t)$, and at the same time $y(t)+u(t)$ presents the first-order differential of the signal. Note that $u(t)$ is inserted both in Eqs. (57) and (58). We also name the above differentiator to be a Lie-group differential algebraic differentiator (LGDAD).

For this example we test the first-order differential of $f(t) = 5t + \sin t$. Under a noise with intensity $\sigma = 0.001$ we obtain a noisy signal $\hat{f}(t_i) = 5t_i + \sin t_i + \sigma R(i)$, where t_i is the sampling time and $R(i)$ is a white noise between $[-1,1]$. By using the LGDAE to solve Eqs. (57)-(59) we fix $h = 0.05$, $\varepsilon_1 = 10^{-4}$ and $\varepsilon_2 = 10^{-10}$, and $\alpha = 1$ is used in Eq. (58). From Fig. 5(a) we can observe that the result of $\dot{f}(t)$ obtained by the LGDAD is much better than that without using u of the original TD solved from Eqs. (55) and (56) with $\alpha = 20$ and $h = 0.001$. It is obvious that the TD has a time-ahead before the given differential signal. The numerical errors are compared in Fig. 5(b), while the control force for the LGDAD is shown in Fig. 5(c). The maximum error of LGDAD is 0.0513, while that for TD is 3.0086.

5.4 The second ITD

Instead of the original third-order Tracking Differentiator (TD) [Han and Wang (1994)]:

$$\dot{x}(t) = y(t), \tag{60}$$

$$\dot{y}(t) = z(t), \tag{61}$$

$$\dot{z}(t) = -\alpha \text{sign}\left(x(t) - f(t) + \frac{y(t)|y(t)|}{2\alpha} + \frac{z(t)|z(t)|}{2\alpha}\right), \tag{62}$$

we propose a new third-order differentiator by the following index-two DAE:

$$\dot{x}(t) = y(t) + u(t), \tag{63}$$

$$\dot{y}(t) = u(t), \tag{64}$$

$$x(t) - f(t) = 0. \tag{65}$$

From the first equation we can obtain the first-order differential by $y(t)+u(t)$, while the second-order differential is given by $\ddot{x}(t) = \dot{y}(t) + \dot{u}(t) = u(t) + \dot{u}(t)$ with the aid of Eq. (64); hence, we further need the following equations to compute $\dot{u}(t)$:

$$\dot{p}(t) = q(t), \tag{66}$$

$$\dot{q}(t) = -\beta \text{sign}\left(p(t) - u(t) + \frac{q(t)|q(t)|}{2\beta}\right). \tag{67}$$

The value $q(t)$ gives $\dot{u}(t)$. The initial values of $p(t)$ and $q(t)$ are set to be 0.1. Here we need to emphasize that Eqs. (63)-(65) are different from Eq. (3) on the

first equation, and also from Eq. (37) on the first and the third equations. Table 1 compares these three type differentiators.

Table 1: The comparisons of three indexed type differentiators of this paper

Index-three differentiator	The first ITD	The second ITD
$\dot{x}=y$	$\dot{x}=y$	$\dot{x}=y+u$
$\dot{y}=u$	$\dot{y}=u$	$\dot{y}=u$
$x-f=0$	$y-g=0$	$x-f=0$

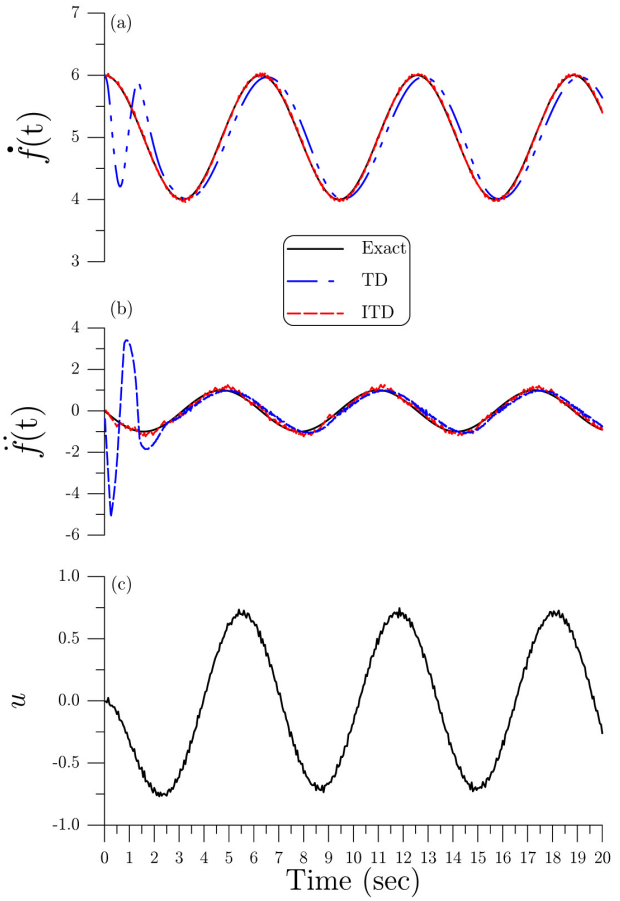

Figure 6: For example of second-order differential under a noise, (a) and (b) comparing the numerical solutions obtained by the second ITD and TD with exact solutions, and (c) the control force for ITD.

By using the LGDAE to solve Eqs. (63)-(65) we fix $\sigma = 0.001$ for noise, $h = 0.05$, $\varepsilon_1 = 10^{-4}$ and $\varepsilon_2 = 10^{-10}$, and $\beta = 1.5$ is used in Eq. (67). From Figs. 6(a) and 6(b) we can observe that the errors of $\dot{f}(t)$ and $\ddot{f}(t)$ obtained by this index-two differentiator (ITD) are much smaller than that without using u of the original TD solved from Eqs. (60)-(62) with $\alpha = 20$ and $h = 0.001$. The maximum errors of ITD are 0.0569 for the first-order differential, and 0.293 for the second-order differential. The control force used in this ITD is shown in Fig. 6(c).

Up to here we have developed two index-two differentiators (ITDs) as shown in Eq. (37) and Eqs. (63)-(65), respectively. When the first ITD is used to compute the first-order differential, the second ITD can compute the first-order differential, which upon together with the TD as shown in Eqs. (66) and (67) can also be used to compute the second-order differential.

Under the same noise $\sigma = 0.001$ and the same stepsize $h = 0.05$, the maximum errors of the first-order differential for the first ITD, the LGDAD modifed from TD and the second ITD are, respectively, 0.06397, 0.0513 and 0.0569. The maximum errors for the LGDAD modified from the second-order RED and the third-order RED are, respectively, 0.09997 and 0.0464. Among these differentiators, the one obtained from the LGDAD modified from the third-order RED provides the most accurate numerical solution of the first-order differential; however, it needs three parameters to be specified. Conversely, the two ITDs are also quite accurate but they do not need any parameter.

5.5 A modification of CHD

Recently, Wang and Lin (2012) have proposed a smooth modification of Eqs. (44) and (45) by multiplying $|x(t) - f(t)|^{\alpha}$ on the right-hand side of Eq. (45). Furthermore, Wang and Lin (2011) proposed the following continuous hybrid differentiator (CHD):

$$\dot{x}(t) = y(t) - k_1|x(t) - f(t)|^{(\alpha+1)/2}\text{sign}(x(t) - f(t)) - k_2(x(t) - f(t)), \tag{68}$$
$$\dot{y}(t) = -k_3|x(t) - f(t)|^{\alpha}\text{sign}(x(t) - f(t)) - k_4(x(t) - f(t)), \tag{69}$$

where $(x(t), y(t))$ is used to track the signal $(f(t), \dot{f}(t))$. As that done in Section 5.2 for the RED and in Section 5.3 for the TD we propose the following modification of CHD:

$$\dot{x}(t) = y(t) - k_1|x(t) - f(t)|^{(\alpha+1)/2}\text{sign}(x(t) - f(t)) - k_2(x(t) - f(t)) + u(t), \tag{70}$$
$$\dot{y}(t) = -k_3|x(t) - f(t)|^{\alpha}\text{sign}(y(t) - \dot{x}(t)) - k_4(x(t) - f(t)), \tag{71}$$

where $u(t)$ is an extra compensated control force, which is solved by the LGDAE to match the algebraic equation (65). This differentiator is the third LGDAD.

We test one case of $f(t) = 2\sin t$ in a time interval of $t \in [0, 50]$ under a large noise $\sigma = 0.01$ being imposed on the given data, where we fix the time sampling rate to be $h = 0.05$. By using the parameters given in Wang and Lin (2011):

$$\alpha = 0.2,\ \ k_1 = 1,\ \ k_2 = 7 \ \text{ if } t \le 1,\ k_2 = 1 \text{ otherwise,}$$
$$k_3 = 8,\ \ k_4 = 25 \ \text{ if } t \le 1,\ k_4 = 8 \text{ otherwise,}$$

we compare the numerical results with the exact solutions in Fig. 7. The maximum error of $f(t)$ is 0.1648, while the maximum error of $\dot{f}(t)$ is 1.138.

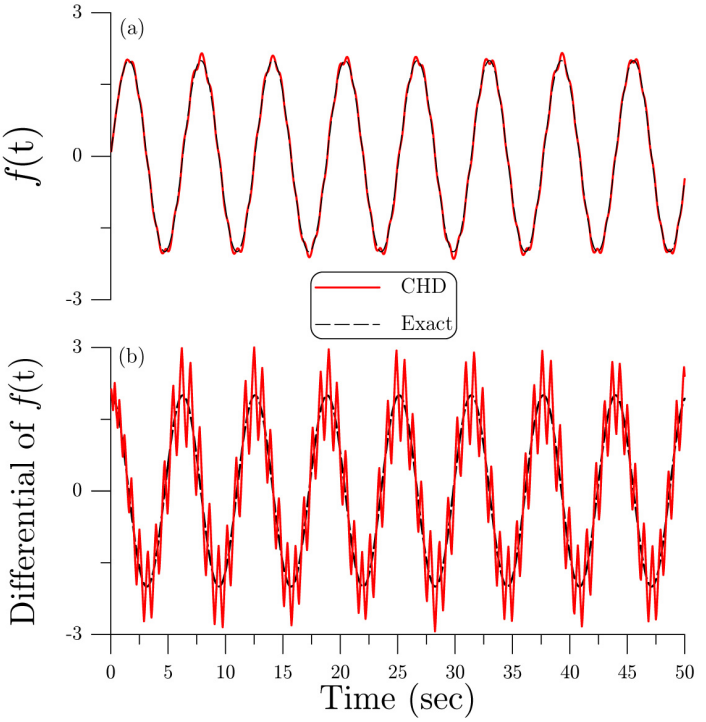

Figure 7: For example of a first-order differential under a noise 0.01, (a) and (b) comparing the numerical solutions obtained by the CHD with exact solutions.

Then we apply the LGDAE to solve the above equations under $\sigma = 0.01$ for noise, $h = 0.05$, $\varepsilon_1 = 10^{-8}$, $\varepsilon_2 = 10^{-12}$, and the same values of the above parameters. From Figs. 8(a) and 8(b) we can observe that the errors of $f(t)$ and $\dot{f}(t)$ obtained by the third LGDAD are much smaller than that without using u of the original CHD solved from Eqs. (68) and (69). The maximum errors of LGDAD are 0.018

for $f(t)$, and 0.588 for $\dot{f}(t)$. When we further filter the above numerical result of $\dot{f}(t)$ by using the CHD in Eqs. (68) and (69) under the following parameters:

$$\alpha = 0.2,\ \ k_1 = 3,\ \ k_2 = 7 \ \text{ if } t \le 1,\ k_2 = 1 \text{ otherwise},\ \ k_3 = 8,\ \ k_4 = 0,$$

the result is improved as shown in Fig. 8(c), where the maximum error is reduced from 0.588 to 0.418. Because the noise is filtered out, the new curve of $\dot{f}(t)$ is more smooth than the original curve in Fig. 8(b).

At the last we compare the tracking effect to the function $f(t) = 1 + 5t + \sin t$ under a noise $\sigma = 0.001$ by using the first index-two differentiator (ITD) in Eq. (37) and that obtained by the CHD. The tracking errors are compared in Fig. 9. When $h = 0.05$ is used in the CHD the errors are far from being comparable with that obtained by the first ITD. So we reduce to $h = 0.01$ for the CHD, and keep $h = 0.05$ for the ITD. However, when the maximum errors for $f(t)$ and $\dot{f}(t)$ are, respectively, 9.99×10^{-4} and 0.06397 for the ITD, they are, respectively, 5.468×10^{-3} and 0.143 for the CHD. Obviously, the first index-two differentiator (ITD) is better than the CHD.

6 Conclusions

The implicit Lie-group $GL(n,\mathbb{R})$ integration algorithm is a new method to effectively and accurately solve nonlinear ODEs. When it is combined with the Newton iterative scheme we can solve the DAEs generated from the real-time numerical differential of noisy signal by using the Lie-group DAE (LGDAE) method. The major contributions in this paper were that we have proposed some modifications of the robust exact differentiator, the tracking differentiator and the continuous hybrid differentiator by suitably adding an extra compensated control force in the governing ODEs and resulting in three Lie-group differential algebraic differentiators (LGDADs), which can eliminate the time-lag or time-ahead and the chattering phenomena, as well as improve the stability and accuracy in the first-order and the second-order differentials of a given noisy signal. The two index-two differentiators (ITDs) as shown, respectively, in Eq. (37) and Eqs. (63)-(65) are simple and without needing of any parameter in the governing ODEs, and when we used them as a differentiator for the first-order differential they provided the most accurate result of the first-order differential than other differentiators. Because the convergence behaviors of the LGDAD and ITD are fast, we have time-saving differentiators to perform the numerical differentials of noisy signal in real time, which can be used in an on-line control algorithm or an on-line identification of time-dependent parameter.

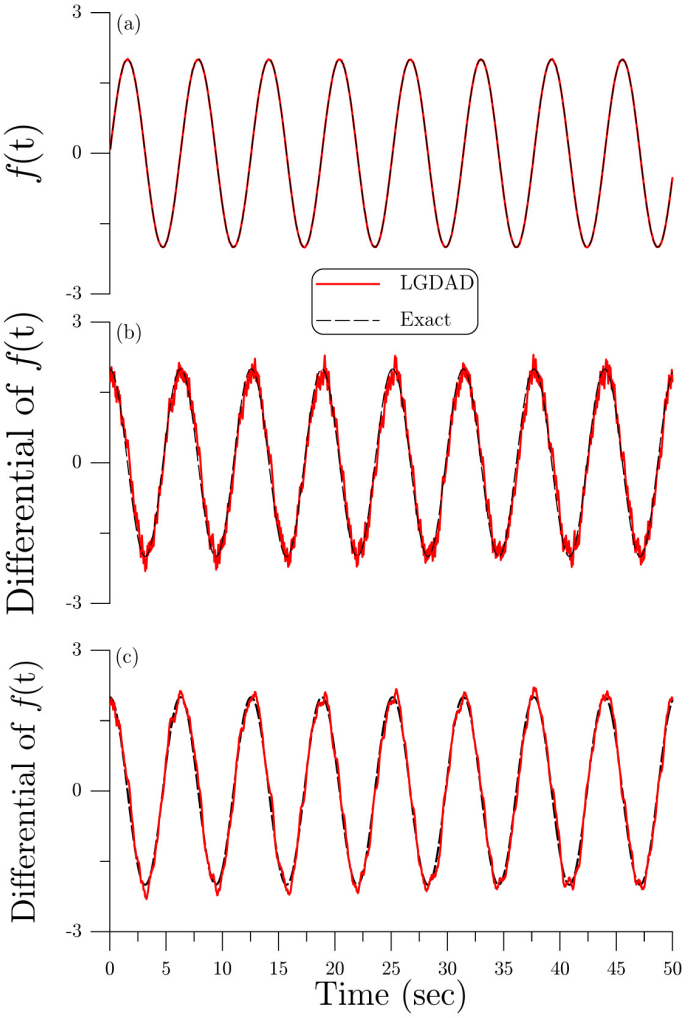

Figure 8: For example of a first-order differential under a noise 0.01, (a) and (b) comparing the numerical solutions obtained by the LGDAD with exact solutions, and (c) a further filtering result.

Acknowledgement

The projects NSC-100-2221-E-002-165-MY3 and NSC-102-2221-E-002-125-MY3 and the 2011 Outstanding Research Award from Taiwan's National Science Council, and the 2011 Taiwan Research Front Award from Thomson Reuters, granted to the first author, are highly appreciated.

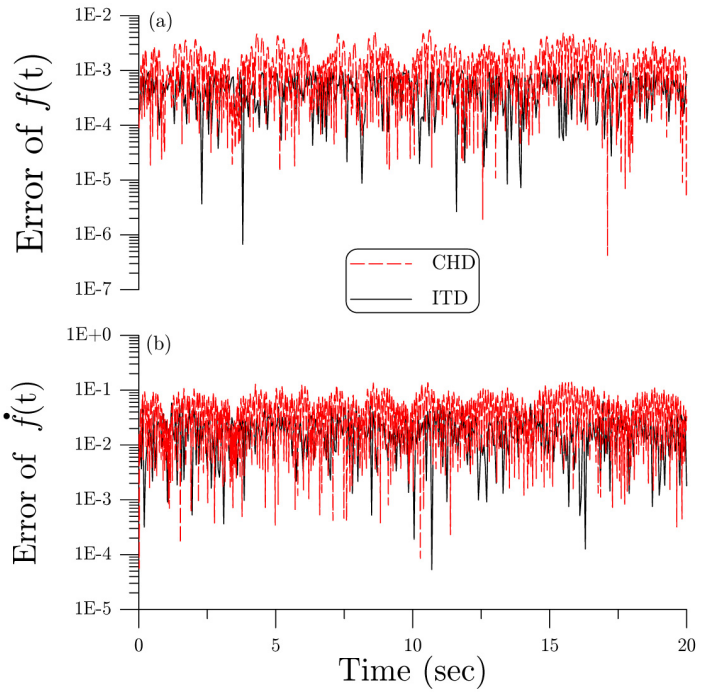

Figure 9: For example of a first-order differential under a noise 0.001, (a) and (b) comparing the numerical errors obtained by the first ITD and that obtained by CHD.

References

Ahn, S.; Choi, U. J.; Ramm, A. G. (2006): A scheme for stable numerical differentiation. *J. Comp. Appl. Math.*, vol. 186, pp. 325-334.

Dabroom, A. M.; Khalil, H. K. (1997): Discrete-time implementation of high-gain observers for numerical differentiation. *Int. J. Contr.*, vol. 72, pp. 1523-1537.

Golembo, B. Z.; Emelyanov, S. V.; Utkin, V. I.; Shubladze, A. M. (1976): Application of piecewise-continuous dynamic systems to filtering problems. *Auto. Remote Contr.*, vol. 37, pp. 369-377.

Guo, B. Z.; Han, J. Q.; Xi, F. B. (2002): Linear tracking-differentiator and application to online estimation of the frequency of a sinusoidal signal with random noise perturbation. *Int. J. Sys. Sci.*, vol. 33, pp. 351-358.

Guo, B. Z.; Zhao, Z. L. (2011a): On convergence of tracking differentiator and application to frequency estimation of sinusoidal signals. *Int. J. Contr.*, vol. 84, pp. 693-701.

Guo, B. Z.; Zhao, Z. L. (2011b): On finite-time stable tracking differentiator without Lipschitz continuity of Lyapunov function. Proc. Chinese Control Conference, 2010, pp. 354-358.

Han, J. Q.; Wang, W. (1994): Nonlinear tracking-differentiator. *J. Sys. Sci. Math. Sci.*, vol. 14, pp. 177-183 (in Chinese).

Ibrir, S. (2004): Linear time-derivative trackers. *Automatica*, vol. 40, pp. 397-405.

Levant, A. (1993): Sliding order and sliding accuracy in sliding mode control. *Int. J. Contr.*, vol. 58, pp. 1247-1263.

Levant, A. (1998): Robust exact differentiation via sliding mode technique. *Automatica*, vol. 34, pp. 379-384.

Levant, A. (2003): Higher-order sliding modes, differentiation and output-feedback control. *Int. J. Contr.*, vol. 76, pp. 924-941.

Levant, A.; Livne, M. (2012): Exact differentiation of signals with unbounded higher derivatives. *IEEE Trans. Auto. Contr.*, vol. 57, pp. 1076-1080.

Liu, C.-S. (2001): Cone of non-linear dynamical system and group preserving schemes. *Int. J. Non-Linear Mech.*, vol. 36, pp. 1047-1068.

Liu, C.-S. (2007): New integrating methods for time-varying linear systems and Lie-group computations. *CMES: Computer Modeling in Engineering & Sciences*, vol. 20, pp. 157-175.

Liu, C.-S. (2013a): A method of Lie-symmetry $GL(n,\mathbb{R})$ for solving non-linear dynamical systems. *Int. J. Non-Linear Mech.*, vol. 52, pp. 85-95.

Liu, C.-S. (2013b): A state feedback controller used to solve an ill-posed linear system by a $GL(n,\mathbb{R})$ iterative algorithm. *Commu. Numer. Anal.*, vol. 2013, Article ID cna-00181, 22 pages.

Liu, C.-S.; Atluri, S. N. (2009): A fictitious time integration method for the numerical solution of the Fredholm integral equation and for numerical differentiation of noisy data, and its relation to the filter theory. *CMES: Computer Modeling in Engineering & Sciences*, vol. 41, pp. 243-261.

Ramm, A.; Smirnova, A. B. (2001): On stable numerical differentiation. *Math. Comp.*, vol. 70, pp. 1131-1153.

Su, Y. X.; Sun, D.; Duan, B. Y. (2005): Design of an enhanced nonlinear PID controller. *Mechatronics*, vol. 15, pp. 1005-1024.

Su, Y. X.; Zheng, C. H.; Mueller, P. C.; Duan, B. Y. (2006): A simple improved velocity estimation for low-speed regions based on position measurements only. *IEEE Trans. Contr. Sys. Tech.*, vol. 14, pp. 937-942.

Tang, Y.; Wu, Y.; Wu, M.; Hu, X.; Shen, L. (2009): Nonlinear tracking-differentiator

for velocity determination using carrier phase measurements. *IEEE J. Selec. Top. Signal Proce.*, vol. 3, pp. 716-725.

Wang, J.; Zhang, J.; Yan, J. (2010): A new second-order nonlinear tracking differentiator and application, 2010 International Conference on Computer Design and Applications (ICCDA), pp. 1318-1322.

Wang, X.; Chen, Z.; Yang, G. (2007): Finite-time-convergent differentiator based on singular perturbation technique. *IEEE Trans. Auto. Contr.*, vol. 52, pp. 1731-1737.

Wang, X.; Lin, H. (2011): Design and analysis of a continuous hybrid differentiator. *IET Contr. Theo. Appl.*, vol. 5, pp. 1321-1334.

Wang, X.; Lin, H. (2012): Design and frequency analysis of continuous finite-time-convergent differentiator. *Aerospa. Tech.*, vol. 18, pp. 69-78.

Wang, X.; Shirinzadeh, B. (2012): Rapid-convergent nonlinear differentiator. *Mech. Sys. Sign. Proces.*, vol. 28, pp. 414-431.

Xue, W.; Huang, Y.; Yang, X. (2010): What kinds of system can be used as tracking-differentiator. Proc. Chinese Control Conference, 2010, pp. 6113-6120.